UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/930,976	08/17/2001	Min-Fu Kao	KAOM3002/EM/7132	1740

23364 7590 08/13/2004

BACON & THOMAS, PLLC
625 SLATERS LANE
FOURTH FLOOR
ALEXANDRIA, VA 22314

EXAMINER	
RAVINDRAN, LATHA	

ART UNIT	PAPER NUMBER
2183	

DATE MAILED: 08/13/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/930,976	KAO ET AL.	
	Examiner	Art Unit	
	Latha Ravindran	2183	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *17 August 2001*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-10* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-10* is/are rejected.

7)☒ Claim(s) *1-3 and 5-8* is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☒ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *17 August 2001* is/are: a)☐ accepted or b)☒ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1 – 10 have been examined. Claims 1 – 10 have been rejected.

Drawings

2. The drawings are objected to under 37 CFR 1.83(a). The drawings must show every feature of the invention specified in the claims. Therefore,

- Claim 3: "...controlling the multiplexer to select one of the instruction group decoding tables based on the group prefix of the compressed instruction, and searching the corresponding original instruction therein by the index of the compressed instruction for being outputted by the multiplexer..."
- Claim 6: "...a second group prefix followed by an op-code index representing a branch condition code, and a displacement index representing a branch target address;..."
- Claim 6: "...the first sub-table being stored with the branch condition codes of the corresponding original instructions, the second sub-table being stored with the branch target addresses of the corresponding original instructions."
- Claim 7: "...op-code index representing an operation code, and an immediate index representing an immediate value..."
- Claim 7: "...the third sub-table being stored with the operation codes of the corresponding original instructions , the fourth sub-table being stored with the immediate values of the corresponding original instructions."
- Claim 8: "...the memory further comprising program codes each consisting of a fourth group prefix followed by an original instruction."

- Claim 9: "...the group prefix is encoded to have a fixed length."
- Claim 10: "...the group prefix is encoded to have a variable length in such a manner that the group prefix of a frequently used instruction is assigned with a relatively short code."

must be shown or the features canceled from the claims. No new matter should be entered.

3. The drawings are objected to because of the following minor informalities:

- What do the dots in Fig. 7 represent? Remove if it is a typo.
- Signal is misspelled as "singal" in Fig. 6, Control Signal Decoder.
- Decoding Execution Unit (65) in Fig. 7 should be Decoding & Execution Unit to match Fig. 6.

4. Corrected drawing sheets are required in reply to the Office action to avoid abandonment of the application. Any amended replacement drawing sheet should include all of the figures appearing on the immediate prior version of the sheet, even if only one figure is being amended. The figure or figure number of an amended drawing should not be labeled as "amended." If a drawing figure is to be canceled, the appropriate figure must be removed from the replacement sheet, and where necessary, the remaining figures must be renumbered and appropriate changes made to the brief description of the several views of the drawings for consistency. Additional replacement sheets may be necessary to show the renumbering of the remaining figures. The replacement sheet(s) should be labeled "Replacement Sheet" in the page header (as per 37 CFR 1.84(c)) so as not to obstruct any portion of the drawing figures. If the

changes are not accepted by the examiner, the applicant will be notified and informed of any required corrective action in the next Office action. The objection to the drawings will not be held in abeyance.

Specification

5. The section headings should appear in upper case, without underlining or bold type. Refer to 37 CFR 1.77.

6. The title of the invention is not descriptive. A new title is required that is clearly indicative of the invention to which the claims are directed.

The following title is suggested: Instruction decompression using instruction group decoding tables.

7. The specification has not been checked to the extent necessary to determine the presence of all possible minor errors. Applicant's cooperation is requested in correcting any errors of which applicant may become aware in the specification.

Claim Objections

8. Claims 1,2,3,5,6,7, and 8 are objected to because of the following informalities:

9. **Claim 1**:

- Line 5 – 6: "...the instructions fetched from memory..."
    - This term is unclear. Do these instructions refer to "compressed instructions" on line 3? If they do, the examiner requests clarification in the claim. If they do not refer to compressed instructions, then the term has a lack of antecedent basis. The examiner notes, that upon further examination, the instructions fetched from memory will be interpreted to

have their antecedent basis in the compressed instructions defined on line 3.

- Line 7: "...an instruction from the memory..."
    - This term is unclear. Is this instruction from memory different from "compressed instructions" (line 3) or "the instructions fetched from memory" (line 5 – 6)? If they refer to prior terms, the examiner requests that "an" is changed to "the" to illustrate antecedent basis in the prior term. The examiner notes, that upon further examination, an instruction from memory will be interpreted to have an antecedent basis to the compressed instructions.
- Line 14 – 15: "...the original instructions of a predetermined type, ..."
    - There is a lack of antecedent basis with this term. Previously, an original instruction (singular) was defined in lines 11 – 12. Now there is a reference to multiple original instructions, which has not been defined before.
- Line 17: "...a corresponding original instruction..."
    - This term is unclear. Is this original instruction different from the original instruction originally defined in lines 11 – 12? If this term refers to the prior term, the examiner requests that "a" is changed to "the" to illustrate antecedent basis in the prior term. The examiner notes, that upon further examination, a corresponding original instruction will be interpreted to have an antecedent basis to the original instruction.

10. **Claim 2**:

- Line 3: "...the original instructions ..."
    - There is a lack of antecedent basis with this term. Previously, an original instruction (singular) was defined in claim 1 on lines 11 – 12. Now, there is a reference to multiple original instructions, which has not been defined before.

11. **Claim 3**:

- Line 8: "...the decoding and execution unit..."
    - There is a lack of antecedent basis with this term. There is no previous mention of a decoding and execution unit.

12. **Claim 5**:

- Line 3: "..a first instruction group decoding table..."
    - This term is unclear. Is this first instruction group decoding table different from the plurality of instruction group decoding tables defined in lines 13 – 14 in claim 1? If this term refers to the prior term, the examiner requests that "a" is changed to "the" to illustrate antecedent basis in the prior term. The examiner notes, that upon further examination, a first instruction group decoding table will be interpreted to have an antecedent basis to the plurality of instruction group decoding tables.
- Line 4: "...the corresponding original instructions."
    - There is a lack of antecedent basis with this term. Previously, a corresponding original instruction was defined in claim 1 in line 17. Now,

there is a reference to multiple corresponding original instructions, which has not been defined before.

13. **Claim 6**:

- Lines 5 – 6: "...a second instruction group decoding table..."
    - This term is unclear. Is this second instruction group decoding table different from the plurality of instruction group decoding tables defined in lines 13 – 14 in claim 1? If this term prefers to the prior term, the examiner requests that "a" is changed to "the" to illustrate antecedent basis in the prior term. The examiner notes, that upon further examination, a second instruction group decoding table will be interpreted to have an antecedent basis to the plurality of instruction group decoding tables.
- Line 7 – 8: "...the branch condition codes..."
    - There is a lack of antecedent basis with this term. Previously, a branch condition code (singular) was defined on line 3. Now, there is a reference to multiple branch condition codes, which has not been defined before.
- Line 8: "...the corresponding original instructions..."
    - There is a lack of antecedent basis with this term. Previously, a corresponding original instruction was defined in claim 1 in line 17. Now, there is a reference to multiple corresponding original instructions, which has not been defined before.
- Line 9: "...the branch target addresses..."

- There is a lack of antecedent basis with this term. Previously, a branch target address (singular) was defined on line 4. Now, there is a reference to multiple branch target addresses, which has not been defined before.

- Line 9 – 10: "...the corresponding original instructions..."
  - There is a lack of antecedent basis with this term. Previously, a corresponding original instruction was defined in claim 1 in line 17. Now, there is a reference to multiple corresponding original instructions, which has not been defined before.

14. **Claim 7**:

- Line 5: "...a third sub-decoding table..."
  - This term is unclear. Is this third sub-decoding table different from the plurality of instruction group decoding tables defined in lines 13 – 14 in claim 1? If this term prefers to the prior term, the examiner requests that "a" is changed to "the" and "sub-decoding" be changed to "instruction group decoding" to illustrate antecedent basis in the prior term. The examiner notes, that upon further examination, a third sub-decoding table will be interpreted to have an antecedent basis to the plurality of instruction group decoding tables.
- Line 7: "...the operation codes..."
  - There is a lack of antecedent basis with this term. Previously, an operation code (singular) was defined on line 3. Now, there is a reference to multiple operation codes, which has not been defined before.

- Line 7: "...the corresponding original instructions..."
    - There is a lack of antecedent basis with this term. Previously, a corresponding original instruction was defined in claim 1 in line 17. Now, there is a reference to multiple corresponding original instructions, which has not been defined before.
- Line 8: "...the immediate values..."
    - There is a lack of antecedent basis with this term. Previously, an immediate value (singular) was defined in line 4. Now, there is a reference to multiple operation codes, which has not been defined before.
- Line 8 – 9 "...the corresponding original instructions..."
    - There is a lack of antecedent basis with this term. Previously, a corresponding original instruction was defined in claim 1 in line 17. Now, there is a reference to multiple corresponding original instructions, which has not been defined before.

15. **Claim 8**:

- Line 3: "...an original instruction..."
    - This term is unclear. If this original instruction different from the original instruction defined in claim 1 in lines 11 – 12? If this term refers to the prior term, the examiner requests that "an" is changed to "the" to illustrate antecedent basis in the prior term. The examiner notes, that upon further examination, an original instruction will be interpreted to have an antecedent basis to original instruction defined in claim 1 in lines 11 – 12.

16. Appropriate correction is required. See MPEP 2173.05(e).

Claim Rejections - 35 USC § 112

17. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

18. Claim 10 is rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

19. The term "frequently used" in claim 10 is a relative term which renders the claim indefinite. The term "frequently used" is not defined by the claim, the specification does not provide a standard for ascertaining the requisite degree, and one of ordinary skill in the art would not be reasonably apprised of the scope of the invention. Frequently used instruction is indefinite. The examiner notes, that upon further examination, frequently used is interpreted as used.

20. The term "relatively short" in claim 10 is a relative term which renders the claim indefinite. The term "relatively short" is not defined by the claim, the specification does not provide a standard for ascertaining the requisite degree, and one of ordinary skill in the art would not be reasonably apprised of the scope of the invention. Relatively short code is indefinite. The examiner notes, that upon further examination, relatively short is interpreted as shorter than a group prefix.

Claim Rejections - 35 USC § 102

21. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

22. Claims 1,2,4,5,8, and 9 rejected under 35 U.S.C. 102(b) as being anticipated by Bealkowski et al. (US Pat. 5,636,352).

Claim 1:

23. Bealkowski et al. disclose a processor comprising:

- a memory (Fig. 1, RAM 14)
- compressed instructions each having a group prefix followed by at least one index; (Fig. 3, Col. 2, lines 41 – 48/ The Condensed Instruction Cell is the compressed instruction. The Condensed Cell Specifier (CCS) is the group prefix. The Instruction Synonyms (IS) is the index.)
- a compressed instruction buffer for storing and buffering the instructions fetched from the memory (Fig. 1, Cache 15, Fig. 8, 32 K-byte Cache)
- a next address logic for sending out a next instruction in the compressed instruction buffer directly; (Fig. 8, Instruction Fetch 820)
- an instruction decompressor for decompressing the compressed instruction sent from the compressed instruction buffer into an original instruction (Fig. 7, Fig. 8, Synonym Management Unit 830, Col. 3, lines 39 – 50, 62 – 67, Col. 4, lines 1 – 3/ The cells are fetched from the 32 K-byte cache (compressed instruction buffer), and sent to the Synonym Management Unit. The Synonym Management Unit is the instruction decompressor. If expansion (decompression) is needed for

a fetched cell (compressed instruction), then it performs the expansion (decompression) to obtain the standard cell (original instruction).)

- where the instruction decompressor has a plurality of instruction group decoding tables, (Fig. 6, Col. 3, lines 24 – 38, lines 62 – 67, Col. 4, lines 1 – 16) / The multi-level Synonym Bank is the plurality of instruction group decoding tables. They are located in the Synonym Management Unit (instruction decompressor).
- each being stored with the original instructions of a predetermined type, and (Col. 3, lines 2 – 4, "Synonym Bank 500 contains the full width standard cell values, entry 1 (520) through entry n (550), each of which corresponds to one Instruction Synonym." Col. 4, lines 4 – 7, "The values of the entries in the synonym bank or banks can be predetermined at the time the microprocessor is designed and possible stored in a read only memory in the microprocessor or hardwired as instruction synonyms."/ The synonym bank is hardwired (stored) with the full width standard cell values, whose value is predetermined at the time the microprocessor is designed. (original instructions of a predetermined type))
- the instruction decompressor selects one of the instruction group decoding tables based on the group prefix of the compressed instruction for searching a corresponding original instruction therein by the index of the compressed instruction. (Fig. 5, Fig. 6, Col. 3/ The plurality of synonym banks are the instruction group decoding tables. (line 24 – 25)They are in the Synonym Management Unit (instruction decompressor). The CCS (group prefix) is used to select the synonym bank. (line 25 – 27) The Instruction Synonym (IS), or

synonym, (index of the compressed instruction) is used to select an entry (search for a corresponding original instruction) in the synonym bank. (line 5 – 6))

Although Bealkowski et al. teach a memory (Fig. 1, RAM 14), Bealkowski et al. is silent about a memory for storing compressed instructions. However, this feature is deemed to be inherent to the Bealkowski et al. apparatus because the 32 K-byte cache stores compressed instructions, which it receives from memory. Therefore, the memory stores compressed instructions. (Fig. 7, Col. 3, lines 39 – 50, 62 – 67, Col. 4, lines 1 – 3, 17 – 25)/ The 32 K-byte cache stores Condensed Instruction Cells (compressed instructions) because the Instruction Fetch fetches instructions, and sends them to the Synonym Management Unit for expansion (decompression) if needed.)

The Bealkowski et al. system would be inoperative if the memory did not store the compressed instructions, as the cache would not have a source of compressed instructions.

Claim 2:

24. The architecture as claimed in claim 1, further comprising a decoding and execution unit (Fig. 8, Issue Logic, IU, BPU, and FPU) including a control signal decoder for decoding the original instructions into control signals (Fig. 8, Issue Logic) and an execution core (Fig. 8, IU, BPU, and FPU) controlled by the control signal decoder (Fig. 8, Issue Logic) for performing corresponding processes.

Claim 4:

25. The architecture as claimed in claim 1, wherein the memory is a read-only memory (ROM). (Fig. 1, ROM 16)

Claim 5:

26. The architecture as claimed in claim 1, wherein the compressed instruction in memory consists of a first group prefix (Fig. 3, CCS 310, Col. 2, lines 42 – 44/ The Condensed Cell Specifier (CCS) is the first group prefix.)

- followed by an instruction index (Col. 2, lines 41 – 48/ A Condensed Instruction Cell is comprised of a CCS and one Instruction Synonyms (IS). The one IS is the instruction index.)
- for searching a first instruction group decoding table stored with the corresponding original instructions. (Col. 3, lines 1 – 10/ The IS (instruction index) is used to select an entry (search) the Synonym Bank (first instruction group decoding table). The Synonym Bank (first instruction group decoding table) is stored with full width standard cell values (corresponding original instructions).

Claim 8:

27. The architecture as claimed in claim 1, wherein the memory further comprises program codes each consisting of a fourth group prefix followed by an original instruction. (Fig. 2, Fig. 7, Col. 2, lines 29 – 40/ The standard instruction cell (program code) consists of an opcode (fourth group prefix) and one or more fields, depending on the instruction format (followed by an original instruction). The standard instruction cell is found in memory because there are instructions that do not need expansion, as illustrated by Fig. 7. The cells are fetched from the cache, which receives its instructions from memory.)

Claim 9:

28. The architecture as claimed in claim 1, wherein the group prefix is encoded to have a fixed length. (Fig. 4, Col. 2, lines 60 - 67/ The CCS (group prefix) is fixed at 8 bits.)

Claim Rejections - 35 USC § 103

29. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

30. Claim 3 is rejected under 35 U.S.C. 103(a) as being unpatentable over Bealkowski et al. (US Pat. 5,636,352) and in further view of Bauer et al. (US Pat. 6,049,862).

Claim 3:

31. Bealkowski et al. teaches the architecture as claimed in claim 1,

- instruction decompressor (Bealkowski et al., Fig. 8, Synonym Management Unit)
- instruction group extractor for extracting the compressed instruction sent from the compressed instruction buffer (Bealkowski et al., Fig. 8, Col. 3, lines 63 – 66/ Synonym Management Unit)
- select one of the instruction group decoding tables based on the group prefix of the compressed instruction, and (Bealkowski et al., Fig. 6, Col. 3, lines 24 - 27/ The plurality of synonym banks are the instruction group decoding tables. The CCS (group prefix) is used to select the synonym bank.)

- searching the corresponding original instruction therein by the index of the compressed instruction (Bealkowski et al., Fig. 5, Col. 3, lines 1 – 15/ The Instruction Synonym (index of the compressed instruction) is used to select an entry (search for a corresponding original instruction) in the synonym bank.)
- outputted to the decoding and execution unit to be executed (Bealkowski et al., Fig. 8/ The Synonym Management Unit is coupled to the Instruction Unit, which has the Issue Logic, which issues the original instructions to their appropriate execution unit.)

32. Bealkowski et al. is silent about the instruction decompressor further including a multiplexer, controlling the multiplexer to select one of the instruction group decoding tables, the [original instruction] being outputted by the multiplexer.

33. Bauer et al. teaches:

- an instruction decompressor further including a multiplexer. (Bauer et al., Fig. 1, Col. 3, lines 57 – 61, Col. 4, lines 1 – 3.)/ Decoder 5 is the instruction decompressor. Mux 10 is the multiplexer.)
- controlling the multiplexer to select one of the instruction group decoding tables (Bauer et al., Fig. 1, Col. 5, lines 23 – 28/ Function unit 9, found in Decoder 5, generates the control signals for Mux 10. This determines which decoding table's (First Decoder 7 or Second Decoder 8) results to output.)
- the [original instruction] being outputted by the multiplexer (Bauer et al., Fig. 1, Col. 3, lines 49 – 54, Col. 5, lines 28 – 35/ The decoded 100-bit-wide program instruction word is the original instruction. Mux 10 outputs the original

instructions, whose source is the decoding tables (First Decoder 7 and Second Decoder 8).)

34. One of ordinary skill in the art would have recognized the advantage of using a multiplexer in Bauer et al. is that the logic design utilizes readily available commercial IC components, and thus its design time is reduced.

35. At the time of the invention, it would have been obvious to a person of ordinary skill in the art to use a multiplexer described by Bauer et al. in the design of the Synonym Management Unit of Bealkowski et al. to determine which synonym bank (instruction group decoding table) to select for output. The CCS field of the Condensed Instruction Cell would provide control inputs to the multiplexer, and determine which Synonym Bank's results to output. (Bealkowski et al. Col. 3, lines 24 – 27).

36. The motivation for doing so is to reduce the design time by using a readily available commercial IC component in Bealkowski et al. This advantage is desirable as the product could be shipped to market sooner.

37. Therefore, it would have been obvious to add the multiplexer described by Bauer et al. to the Synonym Management Unit of Bealkowski et al. to obtain the invention as claimed in Claim 3.

38. Claims 6 and 7 are rejected under 35 U.S.C. 103(a) as being unpatentable over Bealkowski et al. (US Pat. 5,636,352) and in further view of Araujo et al. (Code Compression Based on Operand Factorization).

Claim 6:

39. Bealkowski et al. teaches the architecture as claimed in claim 1. Bealkowski et al. teaches the compressed instruction in memory comprises of a second group prefix. (Bealkowski et al. fig. 3, Col. 2, lines 41 – 48/ The Condensed Instruction Cell is the compressed instruction. The Condensed Cell Specifier (CCS) is the group prefix.)

40. Bealkowski et al. is silent about the compressed instruction in the memory consisting of a second group prefix followed by an op-code index representing a branch condition code, and a displacement index representing a branch target address; the op-code and the displacement indices are used to search a second instruction group decoding table including a first sub-table and a second sub-table, respectively, the first sub-table being stored with the branch condition codes of the corresponding original instructions, the second sub-table being stored with the branch target addresses of the corresponding original instructions.

41. Araujo et al. teaches:

- the compressed instruction in the memory comprising of an op-code index representing an operation code, and a displacement index representing a branch target address. (Araujo et al, Abstract, "The central idea of operand factorization is the separation of program expression trees into sequences of tree-patterns (opcodes) and operand-patterns (registers and immediates). 4. Compression Algorithm, 4.2 Codewords Compaction, 5.4 Branch Target Address, "During decompression the values of addr and offset are retrieved from the IMD dictionary and the branch instruction is assembled."/ The codeword pair is the compressed instruction in memory. Tp is the op-code index representing an

operation code. Op is the displacement index representing a branch target address because the addr and offset are retrieved from the IMD dictionary.)

- the op-code and displacement indices are used to search a second instruction group decoding table including a first sub-table and a second sub-table, respectively (Araujo et al., Figure 5. The Decompression Engine, 5. Decompression Engine, 5.1 Tree-pattern Generation, 5.2 Tree-pattern Generation, 5.3 Immediate Generation/ The second instruction group decoding table is the sub-decoding table is TGEN, TPD, IGEN, IMD. The first sub-table is the tree pattern dictionary (TPD), which stores the opcodes of the instructions. The second sub-table is the IMD, which stores the immediates used by the program. Op, the displacement index, is used to search a second table as it indexes IGEN, and then IMD.)
- the first sub-table being stored with the operation codes of the corresponding original instructions. (Araujo et al. 1. Introduction, "This paper proposes a code compression technique based on the concept of operand factorization. The key idea of this approach is an operation that factors out the operands (operand-patterns) from the expression trees of a program. The factored expression trees are called tree-patterns." 5.1 Tree-pattern Generation, "The Tree-pattern Dictionary (TPD) stores the opcodes encoded by each tree-pattern codeword."/ TPD, the first sub-table, stores the operation codes of the corresponding original instructions by tree-pattern codewords.)

- the second sub-table being stored with the branch target addresses of the corresponding original instructions (Araujo et al., 5.3 Immediate Generation, 5.4 Branch Target Addresses, "During decompression the value of addr and offset are retrieved from the IMD dictionary and the branch instruction is assembled."/IMD, the second sub-table, stores the branch target addresses of the corresponding original instructions, the program.)

42. However, Araujo et al. is silent about:

- the compressed instruction in the memory consists of a second group prefix followed by an op-code index representing a branch condition code
- the first sub-table being stored with the branch condition codes of the corresponding original instructions

43. By stating "the compressed instruction in the memory consists of a second group prefix followed by an op-code index representing a branch condition code", the applicant implies the storage of descriptive material in the compressed instruction. The title "op-code index representing a branch condition code" implies that the compressed instruction has an op-code index field, which represents a branch condition code.

44. By stating "the first sub-table being stored with the branch condition codes of the corresponding original instructions", the applicant implies the storage of descriptive material in the table. The title "the first sub-table being stored with branch condition codes of the corresponding original instructions" implies that the table holds branch condition codes.

45. These differences are only found in the nonfunctional descriptive material and are not functionally involved in the architecture. The implied "op-code index representing a branch condition code" and "branch condition codes" are not analyzed or evaluated by the system in a manner that would alter the operation of the system. No elements are responsive to the particular content of the compressed instruction. No elements are responsive to the particular content of the table before, during, or after the searching the first sub-table. Thus, this descriptive material will not distinguish the claimed invention from the prior art in terms of patentability. See In re Gulack 703 F.2d 1381, 1385, 217 USPQ 401, 404, (Fed. Cir. 1983); In re Lowry, 32 F. 3d 1579, 32 USPQ2d 1031 (Fed Cir. 1994).

46. It would have been obvious to a person of ordinary skill in the art at the time of applicant's invention to store any type of data in the compressed instruction and table because such data does not functionally relate the to architecture and because the subjective interpretation of data does not patentably distinguish the claimed invention.

47. The advantage of operand factorization, the separation of program expression trees into sequences of tree-patterns (opcodes) and operand-patterns (registers and immediates), is the low compression ratio. (Araujo et al., 1. Introduction, "compression ratio = size of compressed program/ size of uncompressed program." 7. Conclusions, "This paper proposes a code compression technique called operand factorization. The best compression ratio using this technique results in a 35% compression ratio.")

48. At the time of the invention, it would have been obvious to one of ordinary skill in the art to implement operand factorization in Bealkowski et al.'s apparatus by modifying

the synonym unit to incorporate operand factorization by creating a Condensed Instruction Cell consisting of a Condensed Cell Specifier (CCS), an opcode index (Tp) to index into an tree pattern dictionary (TPD), and an displacement index (Op) to index into an immediates dictionary (IMD) where the branch target addresses are stored. The compressed instruction (Condensed Instruction Cell) would be recreated by indexing into the dictionaries, and thus, be decompressed.

49. The motivation for implementing operand factorization in Bealkowski et al.'s design is the advantage of reducing the size of the synonym banks. Operand factorization has a compression ratio of 35%. (Araujo et al. 7. Conclusions, "The best compression ratio using this technique results in a 35% compression ratio.") Instead of indexing into a synonym bank and looking up each individual instruction by an Instruction Synonym, an instruction can be recreated through the use of operand factorization. The size of the tables is reduced because the synonym bank does not need to hold every instruction synonym applicable. It just needs to hold the tree pattern dictionary and immediates dictionary.

50. Therefore, it would have been obvious to add operand factorization, described by Araujo et al. to Bealkowski et al.'s design to obtain the invention as specified in claim 6.

Claim 7:

51. Bealkowski et al. teaches the architecture as claimed in claim 1. Bealkowski et al. teaches the compressed instruction in the memory comprises of a third group prefix. (Bealkowski et al. fig. 3, Col. 2, lines 41 – 48/ The Condensed Instruction Cell is the compressed instruction. The Condensed Cell Specifier (CCS) is the group prefix.)

52. Bealkowski et al. is silent about the compressed instruction in the memory consisting of a third group prefix followed by an op-code index representing an operation code, and an immediate index representing an immediate value; the op-code and the immediate indices are used to search a third sub-decoding table including a third sub-table and a fourth sub-table, respectively, the third sub-table being stored with the operation codes of the corresponding original instructions, the fourth sub-table being stored with the immediate values of the corresponding original instructions.

53. Araujo et al. teaches:

- the compressed instruction in the memory comprising of an op-code index representing an operation code, and an immediate index representing an immediate value; (Araujo et al, Abstract, "The central idea of operand factorization is the separation of program expression trees into sequences of tree-patterns (opcodes) and operand-patterns (registers and immediates). 4. Compression Algorithm, 4.2 Codewords Compaction/ The codeword pair is the compressed instruction in memory. Tp is the op-code index representing an operation code. Op is the immediate index representing an immediate value.)
- the op-code and the immediate indices are used to search a third sub-decoding table including a third sub-table and a fourth sub-table, respectively, (Araujo et al., Figure 5. The Decompression Engine, 5. Decompression Engine, 5.1 Tree-pattern Generation, 5.3 Immediate Generation/ The third sub-decoding table is TGEN, TPD, IGEN, IMD. The third sub-table is tree-pattern dictionary (TPD), which stores the opcodes encoded by each tree-pattern dictionary. The fourth sub-table is the IMD,

which stores the immediates used by the program. Tp, the opcode index, is used to search a third sub-decoding table as it indexes TGEN, and then TPD. Op, the immediate index, is used to search a third sub-decoding table as it indexes IGEN, and then IMD.)

- the third sub-table being stored with the operation codes of the corresponding original instructions, (Araujo et al. 1. Introduction, "This paper proposes a code compression technique based on the concept of operand factorization. The key idea of this approach is an operation that factors out the operands (operand-patterns) from the expression trees of a program. The factored expression trees are called tree-patterns." 5.1 Tree-pattern Generation, "The Tree-pattern Dictionary (TPD) stores the opcodes encoded by each tree-pattern codeword."/ TPD, the third sub-table, stores the operation codes of the corresponding original instructions by tree-pattern codewords.)
- the fourth sub-table being stored with the immediate values of the corresponding original instructions. (Araujo et al., 5.3 Immediate Generation, "The IMD module in Figure 5 stores the immediates used by the program."/ IMD, the fourth sub-table, stores the immediate values of the corresponding original instructions, the program.)

54. The advantage of operand factorization, the separation of program expression trees into sequences of tree-patterns (opcodes) and operand-patterns (registers and immediates), is the low compression ratio. (Araujo et al., 1. Introduction, "compression ratio = size of compressed program/ size of uncompressed program." 7. Conclusions,

"This paper proposes a code compression technique called operand factorization. The best compression ratio using this technique results in a 35% compression ratio.")

55. At the time of the invention, it would have been obvious to one of ordinary skill in the art to implement operand factorization in Bealkowski et al.'s apparatus by modifying the synonym unit to incorporate operand factorization by creating a Condensed Instruction Cell consisting of a Condensed Cell Specifier (CCS), an opcode index (Tp) to index into an tree pattern dictionary (TPD), and an immediate index (Op) to index into an immediates dictionary (IMD). The compressed instruction (Condensed Instruction Cell) would be recreated by indexing into the dictionaries, and thus, be decompressed.

56. The motivation for implementing operand factorization in Bealkowski et al.'s design is the advantage of reducing the size of the synonym banks. Operand factorization has a compression ratio of 35%. (Araujo et al. 7. Conclusions, "The best compression ratio using this technique results in a 35% compression ratio.") Instead of indexing into a synonym bank and looking up each individual instruction by an Instruction Synonym, an instruction can be recreated through the use of operand factorization. The size of the tables is reduced because the synonym bank does not need to hold every instruction synonym applicable. It just needs to hold the tree pattern dictionary and immediates dictionary.

57. Therefore, it would have been obvious to add operand factorization, described by Araujo et al. to Bealkowski et al.'s design to obtain the invention as specified in claim 7.

58. Claim 10 is rejected under 35 U.S.C. 103(a) as being unpatentable over Bealkowski et al. (US Pat. 5,636,352), and in further view of Bauer et al. (US Pat.

6,049,862) and IBM (IBM TDB, Opcode Remap And Compression in Hard Wired RISC Microprocessor).

Claim 10:

59. Bealkowski et al. discloses the architecture as claimed in claim 1.

60. However, Bealkowski et al. fails to teach the group prefix is encoded to have a variable length in such a manner that the group prefix of a frequently used instruction is assigned with a relatively short code.

61. Bauer et al. teaches:

- the group prefix is encoded to have a variable length (Bauer et al., Fig. 2, Col. 4, lines 18 – 20, "The division of the total number of bits available in the two fields 11 and 12 is variable and depends on the respective categories of program instructions."/ The First Field 11 is the group prefix.)

62. However, Bauer et al. is silent about the group prefix of a frequently used instruction is assigned with a relatively short code.

63. IBM teaches:

- the group prefix of a frequently used instruction is assigned with a relatively short code.(IBM, Page 1, "A RISC architecture can describe an instruction format with two opcode fields. ... This article describes how the 16 bits of the opcode are remapped to a ten-bit opcode field."/ The opcode field is the group prefix. The opcode indicates a type of instruction, which is used. The opcode, traditionally 16 bits, is mapped to a 10 bit field, which is relatively short code compared to the standard 16 bits.)

64. The advantage of a group prefix of variable length is that it depends on the respective category of program instructions (Bauer et al. Col. 2, lines 18 – 20).

65. The advantage of having the group prefix of a frequently used instruction assigned with a relatively short code is to save decode logic. (IBM, Page 1, "This article describes a way to save decode logic in a RISC microprocessor.)

66. Bealkowski et al.'s CCS field comprises of a primary opcode. (Bealkowski et al., Col. 2, lines 60 – 64). It is fixed at 8 bits. A shorter opcode would enable the CCS field, comprised of the opcode, to be smaller than the fixed 8 bits. In addition, a variable length CCS field, where the CCS field can be smaller than 8 bits, would allow quicker access to the synonym banks. This currently occurs with the design with the variable length Instruction Synonyms where a smaller bit width synonym is used for the most common synonym bank entries. (Bealkowski et al., Col. 3, lines 10 – 15). At the time of the invention, it would have been obvious to a person of ordinary skill in the art to make the CCS field a variable length just like Bauer et al.'s First Field 11 by assigning a shorter opcode to instructions with IBM's opcode remapping scheme.

67. The motivation for doing so would have been creating a faster access time, in the same way a variable length IS has quicker access to the synonym bank. One of ordinary skill in the art would have recognized by creating a variable length CCS field where a shorter field is reserved for commonly used banks, commonly used banks have a shorter access time.

68. Therefore, it would have been obvious to combine the variable length First Field of Bauer et al. and the opcode remapping of IBM to the CCS field of Bealkowski et al. to obtain the invention as specified in claim 10.

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Latha Ravindran whose telephone number is (703)305-8115. The examiner can normally be reached on Monday through Friday 8:30am to 5:00pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on (703) 305-9712. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Latha Ravindran
Examiner
Art Unit 2183

EDDIE CHAN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100